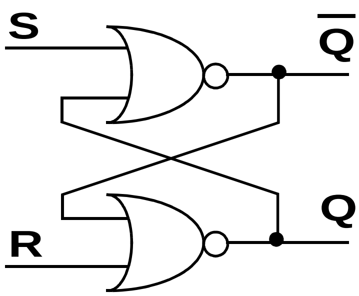
# Set-Reset (SR) Latch

Asynchronous

Level sensitive

cross-coupled Nor gates

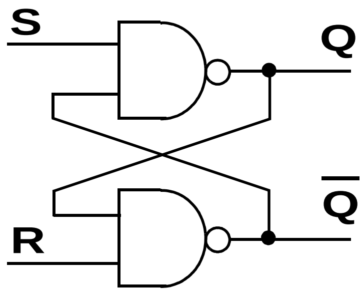
active high inputs (only one can be active)




| S | R | $Q^+$ | $\overline{Q}^+$ | Function |
|---|---|---|---|---|
| 0 | 0 | Q | $\overline{Q}$ | Storage State |
| 0 | 1 | 0 | 1 | Reset |
| 1 | 0 | 1 | 0 | Set |
| 1 | 1 | 0-? | 0-? | Indeterminate State |

cross-coupled Nand gates

active low inputs (only one can be active)



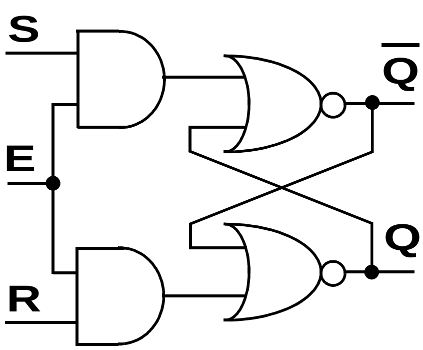

| S | R | $Q^+$ | $\overline{Q}^+$ | Function |
|---|---|---|---|---|
| 0 | 0 | 1-? | 1-? | Indeterminate State |
| 0 | 1 | 1 | 0 | Set |
| 1 | 0 | 0 | 1 | Reset |
| 1 | 1 | Q | $\overline{Q}$ | Storage State |

# Enabled Set-Reset (SR) Latch

Asynchronous

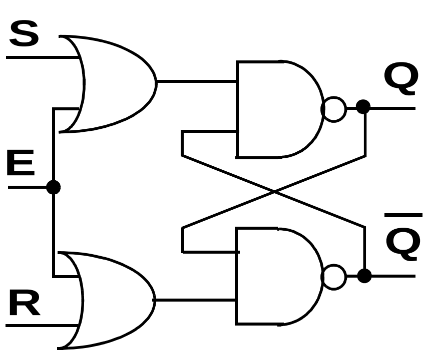
Level sensitive

cross-coupled Nor gates
active high inputs (S & R cannot be active)




| E | S | R | $Q^+$ | $\overline{Q}^+$ | Function |
|---|---|---|---|---|---|
| 0 | x | x | Q | $\overline{Q}$ | Storage State |
| 1 | 0 | 0 | Q | $\overline{Q}$ | Storage State |
| 1 | 0 | 1 | 0 | 1 | Reset |
| 1 | 1 | 0 | 1 | 0 | Set |
| 1 | 1 | 1 | 0-? | 0-? | Indeterminate State |

cross-coupled Nand gates
active low inputs (S & R cannot be active)




| E | S | R | $Q^+$ | $\overline{Q}^+$ | Function |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1-? | 1-? | Indeterminate State |
| 0 | 0 | 1 | 1 | 0 | Set |
| 0 | 1 | 0 | 0 | 1 | Reset |
| 0 | 1 | 1 | Q | $\overline{Q}$ | Storage State |
| 1 | x | x | Q | $\overline{Q}$ | Storage State |

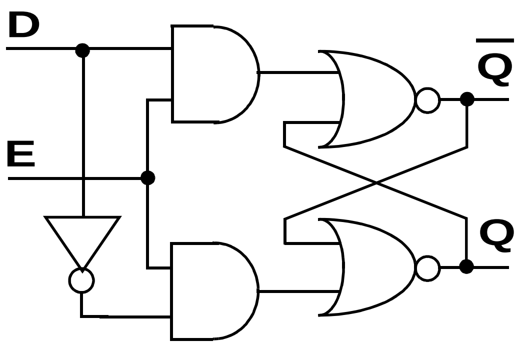
# Transparent D Latch

Asynchronous

Level sensitive

cross-coupled Nor gates

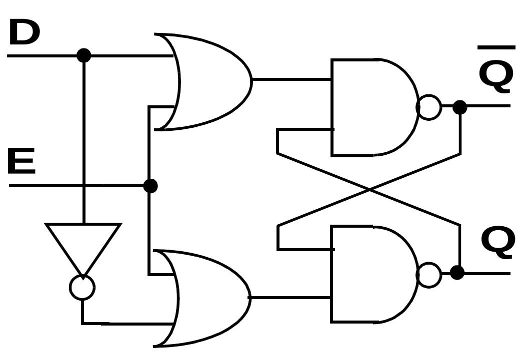
active high enable (E)




| E | D | $Q^+$ | Function |
|---|---|---|---|
| 0 | x | Q | Storage State |
| 1 | 0 | 0 | Transparent Mode |
| 1 | 1 | 1 | Transparent Mode |

cross-coupled Nand gates

active low enable (E)




| E | D | $Q^+$ | Function |
|---|---|---|---|
| 1 | x | Q | Storage State |
| 0 | 0 | 0 | Transparent Mode |
| 0 | 1 | 1 | Transparent Mode |

# D Flip-Flop

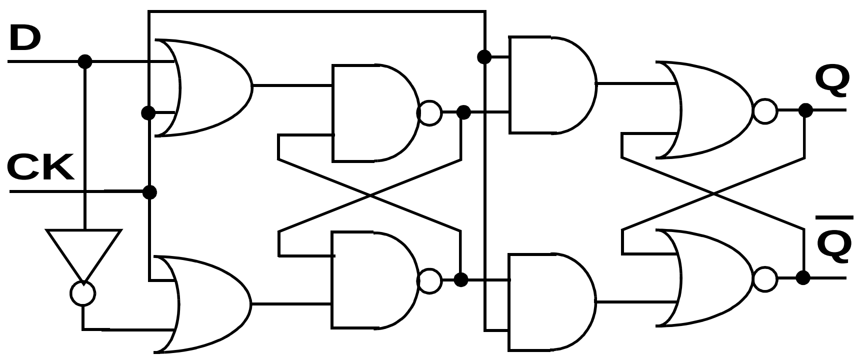
Synchronous (also know as Master-Slave FF)

Edge Triggered (data moves on clock transition)

one latch transparent - the other in storage

active low latch followed by active high latch

positive edge triggered (rising edge of CK)

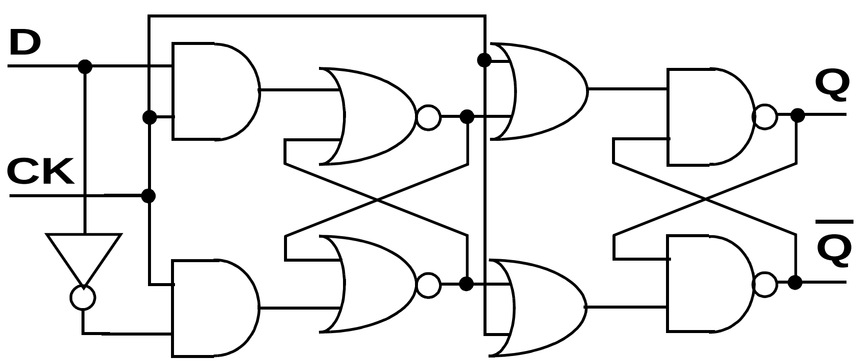



Master Section
active low latch

Slave Section
active high latch

active high latch followed by active low latch

negative edge triggered (falling edge of CK)




Master Section
active high latch

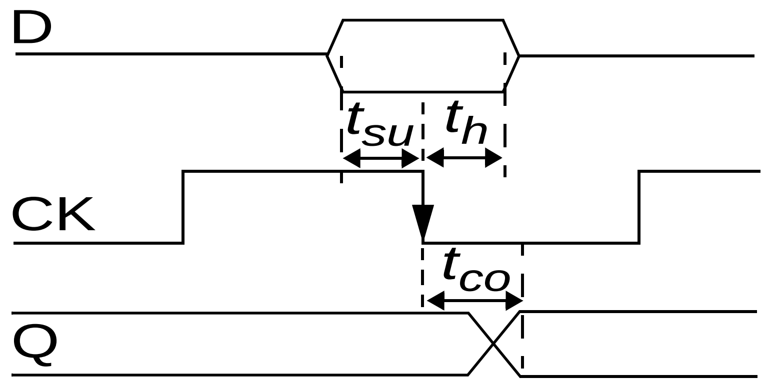
Slave Section
active low latch

# Timing Considerations

Set-up time ($t_{su}$)= minimum time input data must be valid before active edge of clock

Hold time ($t_h$)= minimum time input data must be held valid after active edge of clock

Clock-to-output delay ($t_{co}$)= maximum time before output data is valid with respect to active edge of clock




Set-up *or* Hold Time violation => metastability (Q & $\overline{Q}$ go to intermediate voltage values which are *eventually* resolved to an unknown state)

Set-up & Hold Time violations in a vector set referred to as *clock-data races*

# Timing Considerations

To verify that a sequential logic circuit will work at the specified clock frequency, $f_{clk}$, we must consider the clock period, $T_p$, the propagation delay, $P_{del}$, of the worst case path through the combinational logic, as well as $t_{su}$ and $t_{co}$ of the flip-flops such that the following relationship holds:

For paths from flip-flop outputs to flip-flop inputs:

$$1/f_{clk} = T_p \geq P_{del} + t_{co} + t_{su}$$

For paths from primary inputs to flip-flop inputs:

$$1/f_{clk} = T_p \geq P_{del} + t_{su}$$

For paths from flip-flop outputs to primary outputs:

$$1/f_{clk} = T_p \geq P_{del} + t_{co}$$

For paths from primary inputs to primary outputs:

$$1/f_{clk} = T_p \geq P_{del}$$

Timing analysis and timing simulation CAD tools are typically used for this verification.

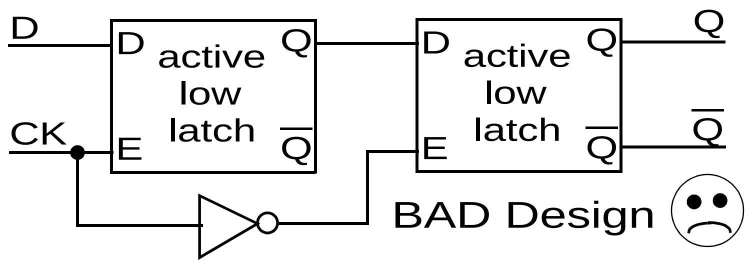
# Good Design Practices

Use single clock, single edge synchronous design techniques as much as possible

Asynchronous interfaces lead to metastability (minimize the async interface & double clock data to reduce probability of metastability)

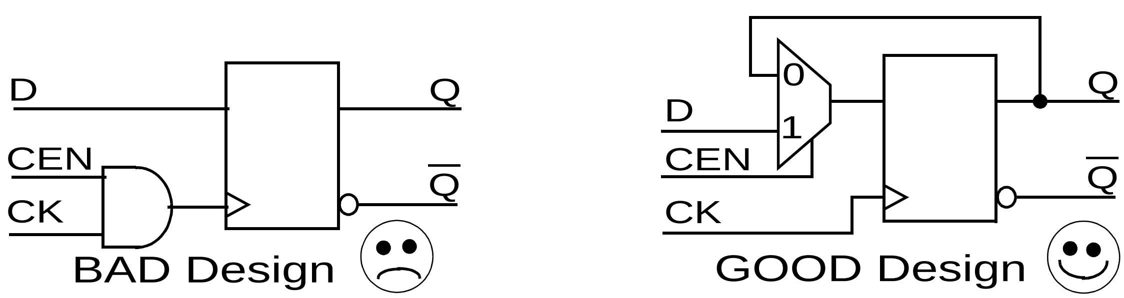
Avoid asynchronous presets & clears on FFs (use sync presets & clears whenever possible)

DO NOT construct a FF from two level sensitive latches of the same type with an inverter on the clock input to one latch




DO NOT gate clocks!!!

Create clock enabled FFs via a MUX to feed back current data




Active high clock enable (CEN)